

events received from the Handler 62. A Mapper 56b maps topology information received from the SAN manager 20 into sub-maps, and a Message Sender 56c is responsible for transmitting, for example, via a socket connection, the topology mapped by the Mapper 56b to the NetView daemon 56 for viewing on the NetView console 52. GTM object wrappers 56d can be utilized

5 by the Mapper 56b for wrapping sub-map objects. The GTM object wrappers 56d further provide helper functions to format GTM API functions into raw String messages to be transmitted by the Sender 56c.

By way of example, FIGURE 39 illustrates an information flow model 364 that depicts the flow of information among the above components when a user, e.g., the operator/administrator,

10 selects a Refresh Topology item from a menu presented on the NetView console 52. In response to such a selection, the NetView console 52 transmits an action event to the NetView Requester 60. The NetView Requester 60 in turn forwards the action event, e.g., as a request, to the Request Handler 62 that processes the request and issues a refresh Topology message to the SAN

15 manager daemon 58.

The manager daemon 58 establishes communication with the manager 20, for example, via an ORB protocol, to retrieve the SAN topology data therefrom. In some embodiments, the manager daemon 58 informs the manger service 38 of the user's security context to allow the manager 38

20 to determine whether the user is eligible for viewing the requested information. In addition, the manager daemon 58 maps the retrieved topology information into sub-maps, and transmits the sub-maps to the NetView daemon 56.

Upon receiving the topology information, the NetView daemon 56 compares the new topology representation with a topology representation previously stored in the NetView database, such as illustrated NetView object database 54a (FIGURE 6), to determine whether the stored topology data requires updating. If the NetView daemon detects changes between the new and the old topology representations, it updates the topology data in the database 54a. The console 52 presents this updated topology information to the user.

A SAN topology model presented to the operator/administrator on the NetView Console 52 may be updated as a result of the operator/administrator's request in a manner described above.

Alternatively, with reference to FIGURE 6, the SAN topology model may require updating when the SAN manager receives reports of topology changes from the discover engine 40. In particular, as discussed above, one or more agents running on one or more hosts connected to the SAN can periodically, or upon request by the discover engine 40 via the query engine 46, obtain information regarding storage devices accessible to their respective hosts, and transmit this information to the discover engine 40. The discover engine 40 collates this information to determine any changes in the SAN topology, and if so, it reports the changes to the SAN manager service 38.

FIGURE 40 presents a flow diagram 366 that schematically depicts the manner in which this new topology data is transmitted from the SAN manager service 38 to the NetView console 52 for presentation to a user, e.g., the SAN administrator.

In particular, upon updating its database, the SAN manager 58 sends a "discover finished" event to the SAN manager daemon 58. The daemon 58 in turn retrieves the new topology information from the SAN manager 38, and maps the topology information into sub-maps, as discussed above. Further, the SAN manager daemon 58 transmits these sub-maps to the SAN NetView
 5 daemon 56. The NetView daemon 56 compares the new topology with a previous topology representation stored in its database to determine whether an update of its topology representation is needed, and if so, it performs the update.

Dynamically Extending File Systems

10 The illustrated embodiment dynamically extends host file systems, without requiring operator intervention and without downtime of the host platform 12. A mechanism for such automatic extension is discussed below. Though discussed in connection with specified file systems, e.g., AIX journal file system, Veritas file system (managed under a Solaris operating system), Unix
 15 file systems (created using Veritas volume manager and managed under a Solaris operating system), it will be appreciated that similar mechanisms can be utilized with hosts operating under other file systems.

Referring back to FIGURE 33, in the illustrated embodiment, when an agent 24 detects that the
 20 utilized portion of a file system associated with a managed host 12 has exceeded a predefined threshold 244, 268 (or upon request of the SAN operator/administrator, or based on other criteria or conditions), it transmits an event notification to the manager 20. The manager 20 determines, based on the predefined policy 240 whether the file system of this managed host 12 should be